

The claims remain pending with no amendment, and listings of claims in the application:

**Listing of Claims:**

1. (previously presented) A digital signature method to be performed by a computer based on braid group conjugacy problem, parameters involved in this method comprising a signatory (S), a signature verifying party (V), a message (M) needing signature, a braid group  $B_n(l)$  divided into a left subgroup  $LB_m(l)$  and a right subgroup  $RB_{n-l-m}(l)$ , an integer  $n$  for a number of generators in the braid group  $B_n(l)$ , an integer  $m$  for a number of generators in a left subgroup  $LB_m(l)$ , an integer  $l$  for an upper bound of a length of a braid, a one way hash function  $h$  from bit sequence  $\{0,1\}^*$  to braid groups  $B_n(l)$ ; said signature method comprising the following steps of:

Step 1. the signatory (S) selecting a braid  $x$  generated from the left subgroup  $LB_m(l)$ , a second braid  $x'$  generated from the braid group  $B_n(l)$ , and a third braid  $a$  generated from the braid group  $B_n(l)$ , by the computer, wherein the computer is adapted to making them meet  $x' = a^{-l}xa$ , moreover, with known  $x$  and  $x'$ , it being impossible to find  $a$  in calculation, and considering braid pair  $(x',x)$  as a public key of signatory (S),  $a$  as a private key of signatory (S);

Step 2. signatory (S) using hash function  $h$  for message (M) needing signature, by the computer, wherein the computer is adapted to get  $y = h(M)$  from the braid group  $B_n(l)$ ;

Step 3. generating a braid  $b$  from the right subgroup  $RB_{n-l-m}(l)$  at random, by the computer, wherein the computer is adapted to signing the message (M) with the private key  $a$  and a generated random braid  $b$  to obtain  $Sign(M) = a^{-l}byb^{-l}a$ ; and

Step 4. the signatory (S) outputting, by the computer, message (M) and a signature of message (M)  $Sign(M)$ .

2. (previously presented) The digital signature method based on braid group conjugacy problem according to claim 1, wherein generating the public key braid pair  $(x',x)$  and the private key braid  $a$  of signatory (S) in said step 1 comprises the following steps of:

Step 1a. selecting a distance  $d$  between system parameter braid groups public key pairs;

Step 1b. representing  $x$  into a left canonical form  $x = \Delta^u \pi_1 \pi_2 \dots \pi_l$ ;

Step 1c. selecting a braid  $b$  at random to belong to a set  $B_n(5l)$

Step 1d. calculating  $x' = b^{-1}xb, a=b$ ;

Step 1e. generating a bit at random, if 1, calculating  $x' = \text{decycling}(x')$ ,  $a = a\pi_i$ ; if not 1, calculating  $x' = \text{cycling}(x')$ ,  $a = a\tau''(\pi_i)$ ;

Step 1f. judging whether  $x'$  belongs to  $SSS(x)$  and whether  $l(x') \leq d$ , if all the conditions are yes, outputting the braid pair  $(x, x')$  as the public key,  $a$  as the private key; if either of them is not, performing step 1e.

3. (previously presented) The digital signature method based on braid group conjugacy problem according to claim 1, wherein the process for obtaining  $y = h(M) \in B_n(l)$  by using the hash function  $h$  in said step 2 comprises the following steps of:

Step 2a, selecting an ordinary hash function  $H$ , with a length of output  $H(M)$  is  $l \lceil \log(2, n!) \rceil$ , then dividing  $H(M)$  into  $l$  sections  $R_1 || R_2 || \dots || R_l$  in equal at one time;

Step 2b, corresponding  $R_i$  to a permutation braid  $A_i$ , then calculating  $h(M) = A_1 * A_2 \dots A_l$ , that is the  $h(M)$  required.

4. (previously presented) The digital signature method based on braid group conjugacy problem according to claim 1, wherein a integer  $n$  for the number of generators in a braid group is in the range of 20~30, an upper value of the braid length is  $l=3$ ,  $d=4$ , and an left subgroup  $n-m=4$ .

5.-7. (canceled)

8. (previously presented) A method for digital signature to be performed by a computer configured to calculate data based on braid groups conjugacy problem and verification thereof, parameters involved in this method comprising a signatory (S), a signature verifying party (V), a message (M) needing signature, a braid group  $B_n(l)$  divided into a left subgroup  $LB_m(l)$  and a right subgroup  $RB_{n-l-m}(l)$ , an integer  $n$  for a number of generators in the braid group  $B_n(l)$ , an integer  $m$  for a number of generators in the left subgroup  $LB_m(l)$ , an integer  $l$  for an upper bound of a length of a braid, a one way hash function  $h$  mapped from bit sequence  $\{0,1\}^*$  to braid groups  $B_n(l)$ ; comprising the following steps of:

Step 1. the signatory (S) selecting a braid  $x$  generated from the left subgroup  $LB_m(l)$ , a second braid  $x'$  generated from the braid group  $B_n(l)$ , and a third braid  $a$  generated from the braid group  $B_n(l)$ , by the computer, wherein the computer is adapted to making them meet

$x' = a^{-1}xa$ , moreover, with known  $x$  and  $x'$ , it is impossible to find  $a$  in calculation, and considering braid pair  $(x', x)$  as a public key of signatory ( $S$ ),  $a$  as a private key of signatory ( $S$ );

Step 2. signatory ( $S$ ) using a hash function  $h$  for message ( $M$ ) needing signature, by the computer, wherein the computer is adapted to get  $y = h(M)$  from the braid group  $B_n(l)$ ;

Step 3. generating a braid  $b$  from the right subgroup  $RB_{n-l-m}(l)$  at random, by the computer, wherein the computer is adapted to signing the message ( $M$ ) with the private key  $a$  and the braid  $b$  generated randomly to obtain  $Sign(M) = a^{-1}byb^{-1}a$ ;

Step 4. the signatory ( $S$ ) outputting, by the computer, the message ( $M$ ) and its signature  $Sign(M)$  to the signature verifying party ( $V$ );

Step 5. the signature verifying party ( $V$ ) obtaining, by the computer, the public key of signatory ( $S$ ) after receiving the message ( $M$ ) and the signature of message ( $M$ )  $Sign(M)$  transmitted from signatory ( $S$ );

Step 6. calculating message  $M$ , by the computer, wherein the computer is adapted to calculate message  $M$  by employing a system parameter hash function  $h$ , to obtain  $y = h(M)$ ;

Step 7. judging whether  $sign(M)$  and  $y$  are conjugate or not, by the computer, wherein the computer is adapted to perform the judging; if not,  $sign(M)$  is an illegal signature, the verification fails; if yes, perform step 8; and

Step 8. calculating  $sign(M)x'$  and  $xy$ , by the computer, wherein the computer is adapted to calculate  $sign(M)x'$  and  $xy$  by using the obtained public key of signatory ( $S$ ), and judging whether they are conjugate or not, if not,  $sign(M)$  is an illegal signature, and the verification fails; if yes,  $sign(M)$  is a legal signature of message ( $M$ ).

9. (previously presented) The method according to claim 8, wherein generating the public key braid pair  $(x', x)$  and private key braid  $a$  of signatory ( $S$ ) in said step 1 comprises the following steps of:

Step 1a. selecting a distance  $d$  between system parameter braid groups public key pair;

Step 1b. representing  $x$  into left canonical form  $x = \Delta^u \pi_1 \pi_2 \dots \pi_l$ ;

Step 1c. selecting a braid  $b$  at random to belong to set  $B_n(5l)$

Step 1d. calculating  $x' = b^{-1}xb$ ,  $a = b$ ;

Step 1e. generating a bit at random, if 1, calculating  $x' = decycling(x')$ ,  $a = a\pi_l$ ; if not 1, calculating  $x' = cycling(x')$ ,  $a = a\tau^u(\pi_l)$ ; and

Step 1f. judging whether  $x'$  belongs to  $SSS(x)$  and whether  $l(x') \leq d$ , if all conditions are yes, outputting the braid pair  $(x, x')$  as the public key,  $a$  as the private key; if either of them is not, performing step 1e.

10. (previously presented) The method according to claim 8, wherein the process for obtaining  $y = h(M) \in B_n(l)$  by using hash function  $h$  in said step 2 comprises the following steps of:

Step 2a. selecting an ordinary hash function  $H$ , with a length of its output  $H(M)$  is  $l$   $[log(2, n!)]$ , then dividing  $H(M)$  into  $l$  sections  $R_1 || R_2 || \dots || R_l$  in equal at one time; and

Step 2b. corresponding  $R_i$  to a permutation braid  $A_i$ , then calculating  $h(M) = A_1 * A_2 \dots A_l$ , that is the  $h(M)$  required.

11. (previously presented) The method according to claim 8, wherein  $n$  for the number of the generation braids in the braid group is in the range of 20~28, an upper value of the braid length is  $l=3$ ,  $d=4$ , and an left subgroup  $n-m=4$ .

12. (previously presented) The method according to claim 8, wherein algorithm  $BCDA$  is employed in judging whether  $sign(M)$  and  $y$  are conjugate or not in step 7 and judging whether  $sign(M) x'$  and  $xy$  are conjugate or not in step 8.

13. (previously presented) The digital signature method based on braid group conjugacy problem according to claim 2, wherein a integer  $n$  for the number of generators in a braid group is in the range of 20~28, an upper value of the braid length is  $l=3$ ,  $d=4$ , and an left subgroup  $n-m=4$ .

14. (previously presented) The digital signature method based on braid group conjugacy problem according to claim 3, wherein a integer  $n$  for the number of generators in a braid group is in the range of 20~28, an upper value of the braid length is  $l=3$ ,  $d=4$ , and an left subgroup  $n-m=4$ .

15. (previously presented) The method according to claim 9, wherein  $n$  for the number of the generation braids in the braid group is in the range of 20~28, an upper value of the braid length is  $l=3$ ,  $d=4$ , and an left subgroup  $n-m=4$ .

16. (previously presented) The method according to claim 10, wherein  $n$  for the number of the generation braids in the braid group is in the range of 20~28, an upper value of the braid length is  $l=3$ ,  $d=4$ , and an left subgroup  $n-m=4$ .